

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Luka Vabič

Optimizacija logistike nabavno prodajnih tokov

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

Ljubljana, 2015

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Luka Vabič

Optimizacija logistike nabavno prodajnih tokov

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: viš. pred. dr. Aljaž Zrnec

Ljubljana, 2015

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Optimizacija logistike nabavno prodajnih tokov

V velikih proizvodnih in storitvenih sistemih že nekaj let prihaja do vpeljave logistike in informacijske podpore le-te. V manjših storitvenih ali proizvodnih sistemih do vpeljave ali optimizacije logistike, zaradi nepoznavanj možnosti optimizacije ali previsokih stroškov implementacije ne prihaja. V diplomski nalogi razvijte prototip sistema, ki bo optimiral logistiko nabavno prodajnih tokov v manjših storitvenih sistemih. Prototip naj na preprost način optimizira proces nabave, koordinacijo z dobavitelji, prodajo, vodenje zalog v realnem času. Zmanjšuje naj tudi pogostost opravljanja inventur.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Luka Vabič sem avtor diplomskega dela z naslovom:

Optimizacija logistike nabavno prodajnih tokov

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom viš. pred. dr. Aljaža Zrneca,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 25. avgusta 2015

Podpis avtorja:

Zahvaljujem se vsem domačim, ki so me čez vsa leta študija tako moralno kot finančno podpirali in me z zgledi ter nasveti usmerjali po pravi poti. Iskrena hvala tudi puncici Ani, ki me je s potrpežljivostjo pospremila čez nastajanje diplomskega dela. Zahvaljujem se tudi mentorju viš. pred. dr. Aljažu Zrnecu za strokovne nasvete in vodenje pri pripravi diplomskega dela.

Kazalo

Povzetek

Abstract

Poglavje 1	Uvod	17
Poglavje 2	Optimizacija logistike s pomočjo informacijsko podprtega sistema	18
Poglavje 3	Razvoj prototipa za optimizacijo logistike nabavno prodajnih tokov.....	20
3.1	Analiza nabavno prodajnega procesa v izbranem podjetju – obstoječe stanje	20
3.2	Možna optimizacija.....	22
3.3	Uporabljena orodja za implementacijo	23
3.3.1	Namestitev razvojnega okolja	23
3.4	Implementacija prototipa za optimizacijo nabavno prodajnih tokov v proizvodnji in logistiki	24
3.4.1	Izdelava ogrodja aplikacije.....	25
3.4.2	Manipulacija s podatki in shramba	26
3.4.3	Modul »blagajna«.....	27
3.4.4	Modul odpis.....	32
3.4.5	Naročilo artiklov.....	34
3.4.6	Modul dobavitelji	39
3.4.7	Vnos zaloge artiklov v sistem.....	41
3.4.8	Modul za pregled zalog in alarmov	42
3.4.9	Vnos novega artikla v sistem.....	44
3.4.10	Dnevniški in delovni zapisi	45
Poglavje 4	Ugotovitve.....	52
4.1	Izzivi ob vpeljavi optimizacije prodajno-nabavnih tokov	54
Poglavje 5	Sklep.....	55

Seznam uporabljenih kratic

Kratica	angleško	slovensko
WMS	warehouse management system	sistem za vodenje skladišč
API	application programming interface	vmesnik
MDI	Multiple document interface	Vmesnik, ki več oken združi pod eno okno

Povzetek

V velikih proizvodnih sistemih že kar nekaj let prihaja do optimizacije procesov. Slednji prinašajo koristi vsem deležnikom proizvodnega procesa, tako delavcem, kot tudi vodstvenemu kadru. Z učinkovito logistiko v proizvodni sistem vstopajo materiali hitreje, točno, brez kopičenja zalog, kar zahteva manjše kapacitete skladišč. Vse to prinaša prihranek podjetju. Iz proizvodnega procesa se proizvodi odpošiljajo hitreje, zaloga do kupcev, grosistov ali maloprodaje, prihaja hitro in pravočasno, kar povečuje zaupanje v našo proizvodnjo.

Optimizirani veliki proizvodni sistemi na dolgi rok prinašajo dobiček, torej se vložek na dolgi rok izplača. Manjši proizvodni sistemi, gostinski obrati in recimo obrtne delavnice, optimizacije procesov ne poznajo, ali se jim obstoječe tehnike optimizacije ne izplačajo.

Za tematiko diplomske naloge smo razvili prototip sistema za optimizacijo nabavno-prodajnega sistema v manjšem gostinskem obratu, lahko pa bi ga aplicirali tudi na kakršenkoli manjši nabavno-prodajni sistem, kjer se zaposleni srečujejo s skladiščenjem materialov oziroma izdelkov in pojmom zaloge.

Ključne besede: optimizacija, skladišče, proizvodnja, storitve, proces, logistika.

Abstract

For the past few years optimization of procedures occurs in numerous larger manufacturing systems. The optimization benefits all the participants of the manufacturing process, not only the executive personnel but also the workers. With efficient logistics the products arrive faster and on time, without the accumulation of stock, which requires smaller capacity of the warehouses. All of these benefits bring savings to the company. The products dispatch faster from the manufacturing processes so the order arrives faster and on time to the buyers – wholesale dealers and retail stores, which increases their trust in the production of the company.

Big optimized production systems bring profit through long time period, so the investment pays off. Smaller production systems such as restaurants, bars and craftsmanships are either not familiar with optimization of processes or these techniques do not pay off for their needs.

A prototype application system for optimization purchase-sale system for smaller restaurants and bars was developed for the topic of diploma. This prototype could also be used for any smaller purchase-sale system where the employees are dealing with storing materials or products.

Keywords: optimization, warehouse, manufacture, process, logistics.

Poglavje 1 Uvod

Namen diplomskega dela " Optimizacija logistike nabavno prodajnih tokov" je razviti prototip sistema, ki optimizira sistem nabave, prodaje ter koordinacije z dobavitelji. Na praktični študiji primera gostinskega obrata z Gorenjske bomo predstavili težave in pomanjkljivosti, ki jih občutijo zaposleni v obratu in s prototipom učinkovito optimizirali storitveni proces obrata.

Do optimizacije proizvodnje dobrin, opravljanja storitev ter logistike prihaja na vseh večjih področjih. Pod optimizacijske metode štejemo tako programske (WMS sistem) kot tudi strojne (avtomatsko vodeno skladišče) rešitve. Te rešitve optimizirajo proizvodne, storitvene in logistične procese tako, da prihranijo čas, denar ter proizvodne in skladiščne kapacitete. Ker je logistika relativno nova veda, se optimizacija procesov manjših podjetij ni dotaknila v tej meri, kot nam dopuščajo najnovejši koncepti in tehnologije. Podjetjem se po navadi prodaja generične rešitve, ki se jih da preprosto aplicirati na večino dejavnosti, vendar se proizvajalci teh rešitev pogosto ne poglobijo dovolj v same proizvodne in storitvene procese, tehnike proizvodnje in vrste storitev, ki jih kupci njihovih rešitev opravljajo. Največkrat gre za sisteme, ki rešujejo le del prodaje, ne optimizirajo pa celotnega poslovnega procesa podjetja.

V prvem delu diplomske naloge smo se poglobili v sam storitveni proces podjetja, ki se ukvarja z gostinstvom. Preučili smo potrebe omenjenega podjetja, katere od njih so že uresničene in potrebe, ki bi morale do sedaj že biti uresničene, pa še niso. Drugi del diplomske naloge opisuje izdelavo rešitve. V njem predstavimo, kako smo rešitev implementirali in katere programske knjižnice smo uporabili. V sklepnem delu diplomske naloge podamo ugotovitve, ali se nam optimizacija izplača. Razviti prototip aplikacije za optimizacijo nabavno-prodajnih tokov v proizvodnji in logistiki je primeren za večino gostinskih obratov, z manjšimi prilagoditvami pa bi ga lahko vpeljali tudi v večino ostalih manjših ali srednje velikih podjetij.

Poglavje 2 Optimizacija logistike s pomočjo informacijsko podprtega sistema

Logistika je bila do osemdesetih let prejšnjega stoletja pojmovana kot vodenje skladiščenja in transporta, v devetdesetih letih pa se je njen obseg razširil in zajel celotno oskrbno verigo. Sedaj pojem vsebuje vse materialne tokove – tok materiala v podjetje, tok v proizvodno-storitvenem procesu in tok izdelkov do končnih strank [1] .

Poznamo 5 stopenj vodenja materialnega toka [1]:

- načrtovanje,
- oskrbovanje,
- skladiščenje,
- proizvodnja,
- distribucija.

Ker je logistika v bistvu povezovalni proces, katere namen je optimizacija toka materialov, polizdelkov in izdelkov skozi podjetje do uporabnika, se funkcija logistike širi iz samega podjetja k dobaviteljem in tudi h končnem uporabniku produkta ali storitve. To je bistvo vodenja oskrbne verige, ki ga podjetje uresničuje z učinkovito razvitim logističnim sistemom [1].

Za učinkovito razvit in informacijsko podprt logistični sistem, ki izboljšuje tokove med dobavitelji, podjetjem in končnim uporabnikom izdelka ali storitve, so pomembne logistične odločitve na štirih ravneh.

Raven odločanja	Področja odločanja
Operativna raven (stalno)	Obdelava naročil, stanje zaloge, spremljanje izvršitve naročil.
Kontrola procesov (večkrat dnevno)	Nadzor nad prevzemom in izdajo izdelkov v skladišče.
Taktično izvajanje (pogosto)	Spremljanje gibanja zaloge.
Strateško načrtovanje (letno)	Organizacija logistike, ocenjevanje doseženih rezultatov na področju logistike.

[1]

V storitvenem procesu, ki smo ga optimizirali v diplomski nalogi, nastopata dva člena v oskrbni verigi. To sta gostinski lokal, katerega storitveni proces smo analizirali in dobavitelji artiklov. V naslednjem poglavju bomo ob analizi prodajno-nabavnega procesa ugotovili, da je proces mogoče optimizirati. Stremeli bomo k zadovoljitvi štirih logističnih odločitev z implementacijo prototipa, ki bo omogočal obdelavo naročil, spremljanje zaloge in njenega gibanja in nadzor nad prevzemom artiklov. Hkrati si bomo prizadevali za čim večjo avtomatizacijo procesov in avtomatizacijo poslovanja med samimi členi oskrbne verige.

Cilj diplomske naloge je razvoj učinkovitega sistema za optimizacijo prodajno-nabavnih tokov, kar bomo poskušali uresničiti z razvojem informacijsko podprtega logističnega sistema. Sistem bo obsegal module za načrtovanje nabave, oskrbo gostinskega lokala z artikli, skladiščenje artiklov in beleženje zaloge v realnem času v skladišču in distribucijo artiklov strankam. S tem bomo zadostili štirim od petih zgoraj navedenim stopnjam vodenja materialnega toka. Proizvodnje artiklov v storitvenem procesu, ki smo ga analizirali, ne beležimo, zato te stopnje vodenja materialnega toka ne bomo implementirali.

Poglavje 3 Razvoj prototipa za optimizacijo logistike nabavno prodajnih tokov

3.1 Analiza nabavno prodajnega procesa v izbranem podjetju – obstoječe stanje

Analizo smo izvedli v podjetju, ki se ukvarja z gostinsko dejavnostjo. Gre za bar, ki poleg pijače ponuja tudi prigrizke in pice. Višek obiskov beleži med koncem tedna, natančneje v soboto. Povečan obisk beležijo tudi ob petkih. Podjetje prodaja večinoma pijače, prednjači pivo, točeno in ustekleničeno.

Nabavo opravljajo sproti, zaloge v realnem času ne beležijo. Beležijo nakupljene artikle, ter vsakotedensko opravijo inventuro, kjer preverijo, če dejansko število artiklov ustreza razliki med številom nabavljenih artiklov in številu prodanih artiklov. Za nakup artikla se odločijo takrat, ko zaposleni ob iznosu artikla iz skladišča ugotovi, da artikla primanjkuje. Artikla prične primanjkovati takrat, ko količina artikla v skladišču pade pod določeno vrednost, ki je odvisna od artikla samega. Dobaviteljev ne menjujejo, ampak so ustaljeni. Dobavitelje predstavljajo trgovine na debelo.

V lokalu se lahko dogajajo tudi nesreče, kar pomeni, da se lahko kakšen artikel razbije, razlije, lahko ga ukradejo ali ga stranka zaradi nezadovoljstva z opravljeno storitvijo reklamira. V slednjih primerih zaposleni odpise artikel, kar se zabeleži v posebnem zvezku.

Sestavljeni artikli so izdelki, ki jih sestavlja več artiklov. Večinoma gre za izdelke z mlekom (bela kava, ki jo sestavljata mleko in kava) in mešane alkoholne pijače (rum-cola, ki jo sestavljata Coca Cola in rum) in podobno.

Ob odpisu sestavljenih artiklov vodja lokala ob inventuri ne pozna točne količine odpisa artiklov, ki sestavljajo sestavljene artikle. V primeru odpisa zaradi razbitja bele kave vodja lokala ne ve, za koliko se je zmanjšala zaloge mleka in za koliko zaloge kave ter zakaj je do odpisa sploh prišlo.

Ob analizi storitvenega procesa lokala smo ugotovili, da inventuro opravljajo vsak teden, ravno zaradi neoptimalno organiziranega procesa odpisov. Le z vsakotedensko inventuro lahko usklajujejo število artiklov v skladišču in število artiklov v blagajniškem sistemu.

Ugotovili smo, da je inventura zamudna zaradi ugotavljanja zaloge artiklov, ki jih ne prodajajo po kosu, ampak so artikli skladiščeni v različnih količinah npr. po kilogramih ali litrih (mleko, kava, žganje). Tekočinsko količino preverijo s tehtanjem. Stehtajo embalažo, nato težo pretvorijo v želeno količino s pomočjo razpredelnice. Le-ta z upoštevanjem gostote tehtanega artikla prikaže pravilno tekočinsko količino (koliko litrov je 40 dag mleka). Pri artiklih, katerih količina se izraža v utežnih merah, pretvarjanje ni potrebno.

Inventura se opravlja vsak teden zaradi neučinkovitega beleženja odpisov, ki povzročajo netočno beleženje pravilne zaloge v realnem času. Želja zaposlenih v lokalu je, da bi obstajal sistem za bolj učinkovito beleženje odpisov in vodenje zaloge v realnem času, saj bi to zmanjšalo število potrebnih inventur.

Izračunajmo čas, ki ga vodja lokala in zaposleni potrebujejo za spremljanje odpisa, preverjanje manjkov v skladišču, opravljanje inventure ter naročanje artiklov.

Gre za preprost izračun:

$$4 \times (3 + \widetilde{3.5} + 1 + 1) = \frac{\widetilde{34}}{4} = \widetilde{8.5}$$

Številke nam predstavljajo:

- 4 tedni v mesecu,
- 3 ure na teden za odhod do dobavitelja in nakup artiklov v trgovini na debelo,
- približno 3.5 ure za tedensko inventuro,
- 1 ura za vnos artiklov v sistem in blagajniški sistem,
- 1 ura za urejanje odpisov.

Ugotovimo, da se samo za urejanje stanja v skladišču, preverjanje prodaje in vnos artiklov porabi več kot en delovni dan na teden.

3.2 Možna optimizacija

Ob analizi poslovnega procesa v poglavju 3.1 smo ugotovili, da bi bilo z učinkovito programsko rešitvijo mogoče vpeljati naslednje izboljšave:

- pospešiti proces prodaje artiklov,
- skrajšati čas naročanja artiklov pri dobaviteljih,
- zmanjšati število inventur,
- zaposlenim omogočiti pregled stanja zalog v realnem času,
- izboljšati in poenostaviti način odpisa artiklov,
- omogočiti pregled in filtriranje dnevniških zapisov transakcij v prototipu aplikacije,
- omogočiti izvoz dnevniških podatkov v formatu .xlsx.

Ob odločitvi za implementacijo se zavedamo naslednjih težav:

- ob nedoslednem beleženju odpisanih artiklov lahko prihaja do neskladja z zalogo artiklov v skladišču in stanjem v samem sistemu,
- sistem ne predvideva zlorab sistema s strani zaposlenih (kraja artiklov s skladišča, goljufije pri prodaji artiklov),
- proces inventure ne bo avtomatiziran. Zalogo artiklov v skladišču bo na določen časovni interval še vedno potrebno ročno preveriti.

Proces nabave in prodaje bo potekal pol avtomatizirano. Zaposleni bodo prodajo in odpis izdelkov, vnos novih izdelkov in vnos nove zaloge v sistem vnašali ročno. Sistem bo omogočal avtomatsko dnevniško beleženje in avtomatsko naročilo izdelka ob padcu zaloge izdelka pod minimalno dovoljeno zalogo.

3.3 Uporabljena orodja za implementacijo

Za implementacijo prototipa smo uporabili programski jezik Java.

Java je sodoben programski jezik, katerega prvi začetki segajo v leto 1991. Je eden najpomembnejših programskih jezikov, ki ga odlikujejo preprostost, objektna orientiranost, robustnost, neodvisnost od sistema, varnost in visoka zmogljivost [2].

Razlog za uporabo Jave je preprosta uporaba in stabilnost samega programskega jezika, dobra dokumentacija, varnost in možnost dobrega povezovanja z zalednimi sistemi, v tem primeru s podatkovno bazo MySQL. Hkrati se v gostinstvu večinoma še vedno uporabljajo namizni računalniki in je zato izdelava namizne aplikacije bolj primerna kot izdelava spletne aplikacije.

Za shrambo podatkov smo uporabili podatkovno bazo MySQL. Gre za popularno odprtokodno implementacijo relacijske podatkovne baze, ki za delo s podatki uporablja jezik SQL [3]. Uporaba in namestitev podatkovne baze MySQL je lahka, v veliki meri ni plačljiva, ker pa se MySQL uporablja že dolgo časa, za pomoč obstaja velika skupnost (Community), ki stoji v ozadju.

Prednosti uporabe MySQL so: visoka zmogljivost sistema, zanesljivost, enostavno upravljanje s samo podatkovno bazo, možnost uporabe podatkovne baze na več platformah, s hitro vzpostavitvijo sistema brez prilagoditev. Prav tako pa se podatkovna baza MySQL enostavno lahko uporabi tudi pri razvijanju aplikacij v drugih programskih jezikih, kot so PHP, VisualBasic, .NET [4].

3.3.1 Namestitev razvojnega okolja

Za razvoj smo uporabili standardni prenosni računalnik, na katerega smo namestili programsko okolje JAVA(TM) SE Runtime Environment (build 1.8.0_25-b18) [5], virtualni paket XAMPP [6], preko katerega smo dostopali do podatkovne baze MySQL (Server version: 5.6.20 - MySQL Community Server (GPL)). Prototip smo razvili v okolju NetBeans 8.01 [7], podatkovno bazo pa implementirali s pomočjo uporabniškega vmesnika v PHPmyadmin [8].

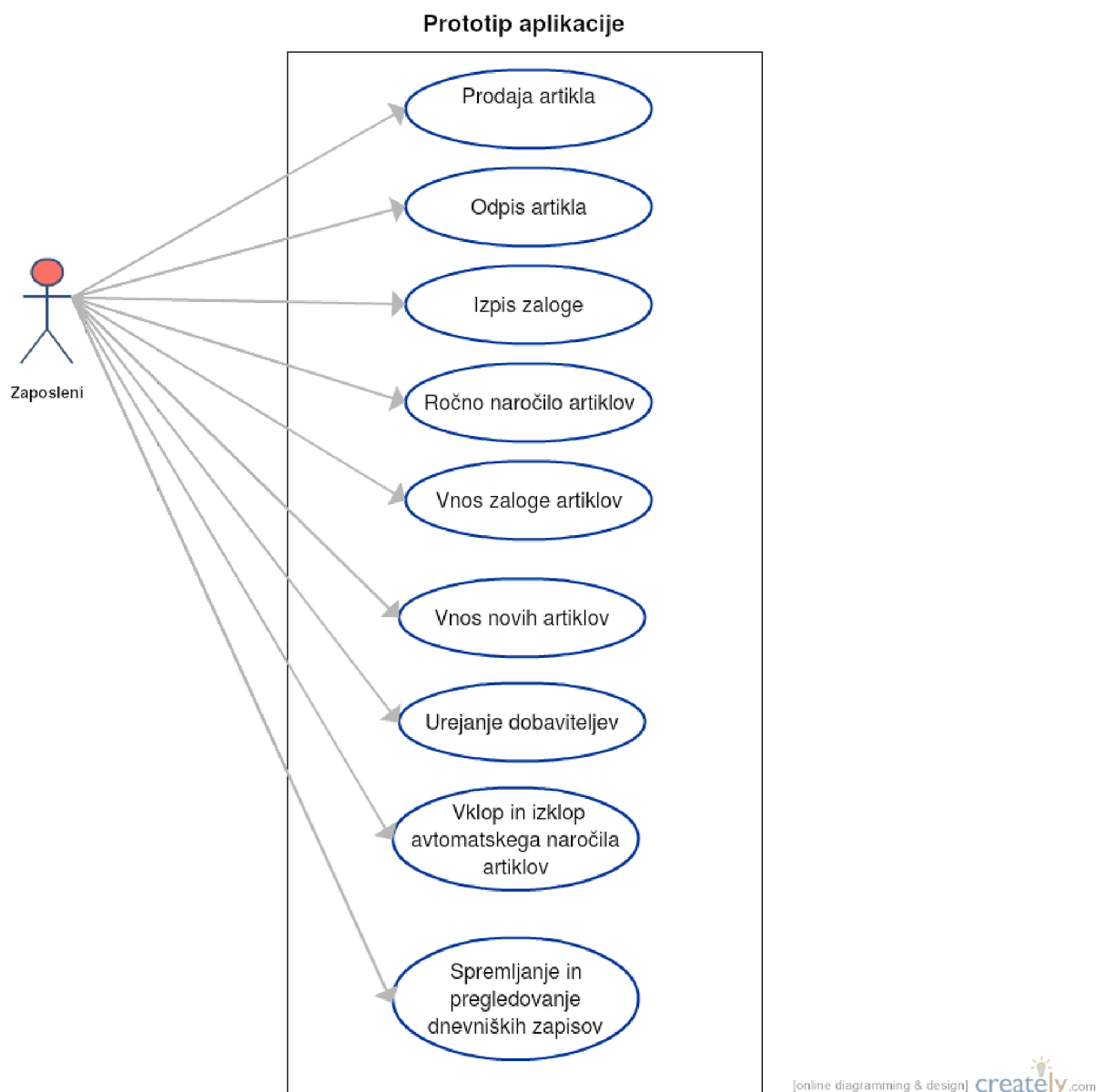
3.4 Implementacija prototipa za optimizacijo nabavno prodajnih tokov v proizvodnji in logistiki

V prototipu smo razvili naslednje funkcionalnosti:

- modul za prodajo artiklov,
- modul za odpis artiklov,
- modul za spremljanje stanja zaloge artiklov v realnem času,
- modul za ročno naročanje artiklov,
- razred za avtomatsko naročanje artiklov pri samih dobaviteljih,
- modul za pregledovanje in izvoz dnevniških zapisov,
- modul za vnos novih artiklov,
- modul za vnos zaloge artiklov,
- modul za pregledovanje in urejanje dobaviteljev artiklov.

Funkcionalnosti sistema lahko prikažemo z diagramom primera uporabe, ki prikazuje funkcionalnosti in akterje v interakciji s sistemom.

Slika 1 prikazuje diagram primera uporabe za gostinski lokal, katerega proces smo analizirali.



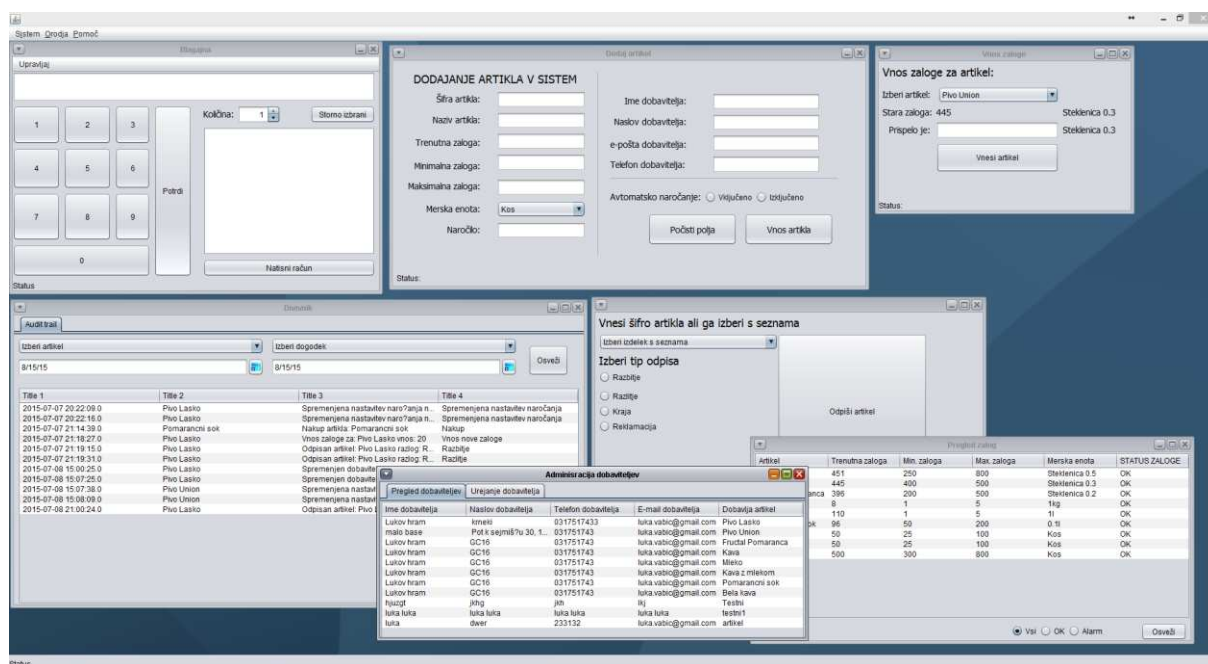
Slika 1: Skica funkcionalnosti sistema in akterji v interakciji z njim.

3.4.1 Izdelava ogrodja aplikacije

Za izdelavo aplikacije smo uporabili JDesktopPane, to je vmesnik, ki več oken združi v eno okno. Upravlja vse potencialno prekrivajoče se notranje okvirje[9] [10]. Tako smo zagotovili, da bo dodajanje modulov, ki jih bomo predstavili kot JInternalFrame preprosto in znotraj same aplikacije. JInternalFrame je lahki objekt, ki ponuja številne funkcije okvirjem, ki mu jih lahko dodamo (JDesktopPane) kot so zapiranje, vlečenje po zaslonu, minimiziranje itd.

Skupek vseh modulov bo prikazan v enem oknu, posamezni moduli pa kot podokna, ki jih bomo lahko odpirali, zapirali in minimizirali. S tem bomo zagotovili prijetno uporabniško izkušnjo ter enotno grafično podobo aplikacije.

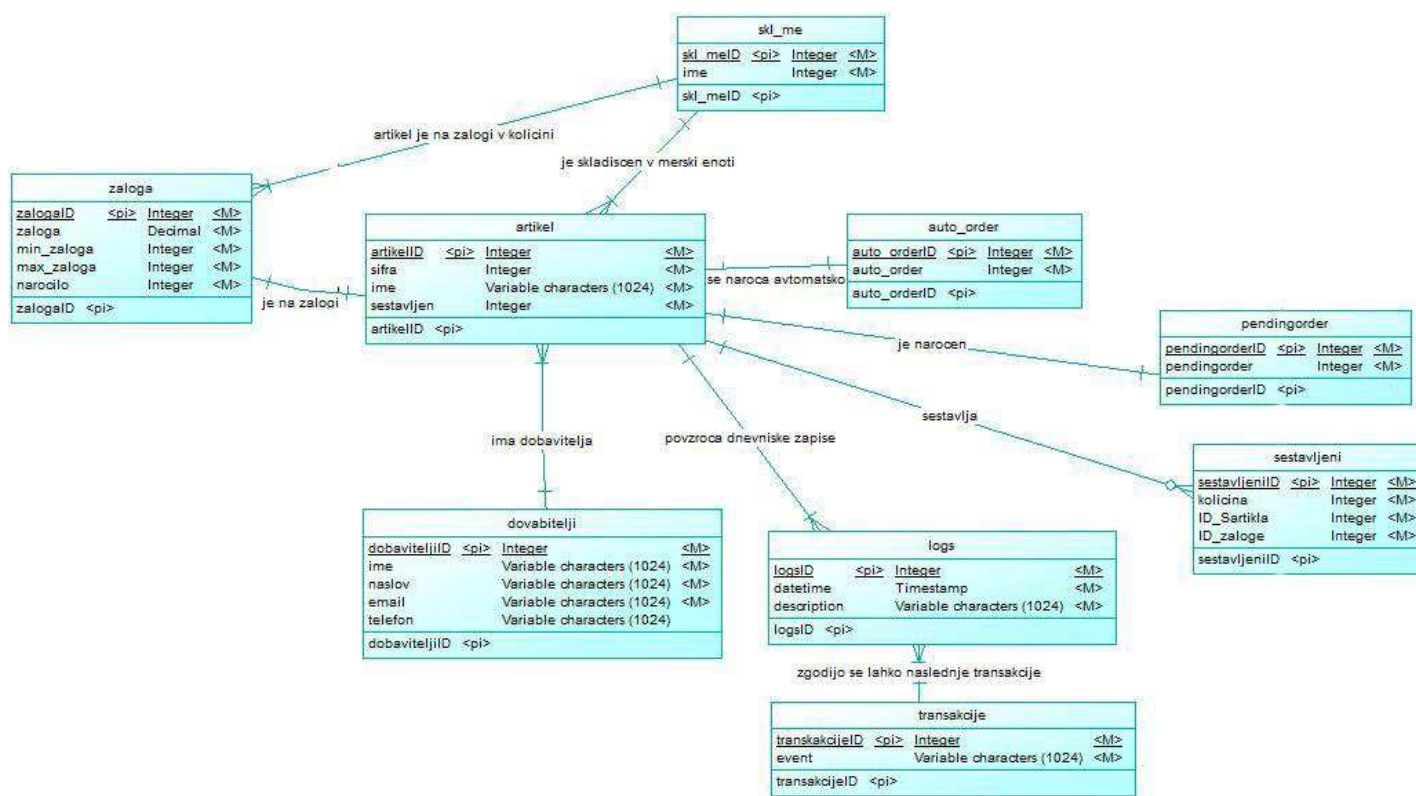
Slika 2 prikazuje zaslonsko masko razvitega prototipa aplikacije z vsemi odprtimi moduli.



Slika 2: Prikaz zaslonske maske razvitega prototipa aplikacije z vsemi odprtimi moduli.

3.4.2 Manipulacija s podatki in shramba

Implementirali smo podatkovno bazo, ki služi za shrambo in manipulacijo s podatki. Slika 3 prikazuje tabele podatkovne baze prototipa, povezave med tabelami in primarne ključe tabel.



Slika 3: Prikaz tabel podatkovne baze, povezav med njimi in primarni ključi tabel.

Za manipulacijo s podatkovno bazo smo izdelali razred Dbconnect. Pri tem smo se upirali na dokumentacijo gonilnika JDBC, ki opiše postopek povezovanja na podatkovno bazo MySQL s programskim jezikom Java. [11]. Razred Dbconnect vsebuje metode, ki jih kličemo iz razredov, ki implementirajo JInternalFrame in predstavljajo module aplikacije – blagajna, dobavitelji, odpisi... Razred s pomočjo knjižnice mysql-connector-java vzpostavi povezavo na podatkovno bazo in pridobiva podatke, vstavlja nove podatke ali jih le posodablja. Vsaka metoda v primeru nezmožnosti manipulacije s podatki sproži izjemo.

3.4.3 Modul »blagajna«

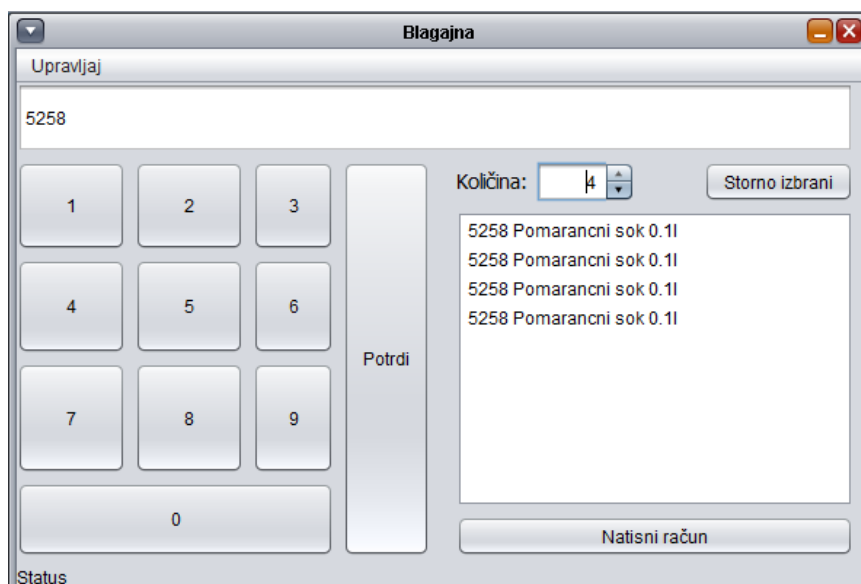
Za prodajo artiklov potrebujemo blagajno. Do sedaj se je izbira artiklov za prodajo v blagajni izvajala ročno. Vzemimo za primer Fanta 0.2 l. V sodobnih blagajniških sistemih je prodaja tega artikla zamudna, ker je med vsemi pijačami potrebno izbrati brezalkoholno pijačo. Nato izmed brezalkoholnih pijač izbrati pravilen artikel in potrditi nakup. V našem primeru je s

čitalnikom črtnih kod potrebno prebrati črtno kodo artikla in ga dodati na seznam za prodajo. Ob potrditvi nakupa se zaloga artiklov zmanjša za število prodanih artiklov. Tabela 1 prikazuje zmanjšanje zaloge artikla Fanta 0.2l za en kos, ob prodaji enega kosa artikla.

Artikel	Stara zaloga	Nova zaloga
Fanta 0.2l	100	99

Tabela 1: Prikaz zmanjšanja zaloge za en kos artikla Fanta 0.2l v podatkovni bazi prototipa aplikacije.

Slika 3 prikazuje modul blagajne, ki služi za prikaz prodaje artiklov. Šifro artikla preberemo z bralnikom črtnih kod. Lahko jo vpišemo ročno. Artikel je mogoče s seznama za prodajo izbrisati z gumbom *storno izbrani*.



Slika 4: Prikaz modula *blagajna* na prototipu aplikacije, ki služi za prikaz prodaje artiklov.

3.4.3.1 Prodaja sestavljenih artiklov

Prodajo sestavljenih artiklov smo implementirali tako, da je beleženje zaloge v realnem času še vedno pravilno. Slika 5 prikazuje tabelo *sestavljene*. Opazimo, da sta sestavljena artikla z *ID_Sartikla* 6 in 8, sestavljena iz artiklov z *ID_artikla* 4 in 5, katerima se zaloga zmanjša za 0.15 in 0.015 merske enote, ki je definirana v tabeli *skl_me*.

	ID	ID_Sartikla	ID_artikla	ID_zaloge	kolicina
Edit Copy Delete	1	6	4	4	0.15
Edit Copy Delete	2	6	5	5	5
Edit Copy Delete	3	8	4	4	0.015
Edit Copy Delete	4	8	5	5	5

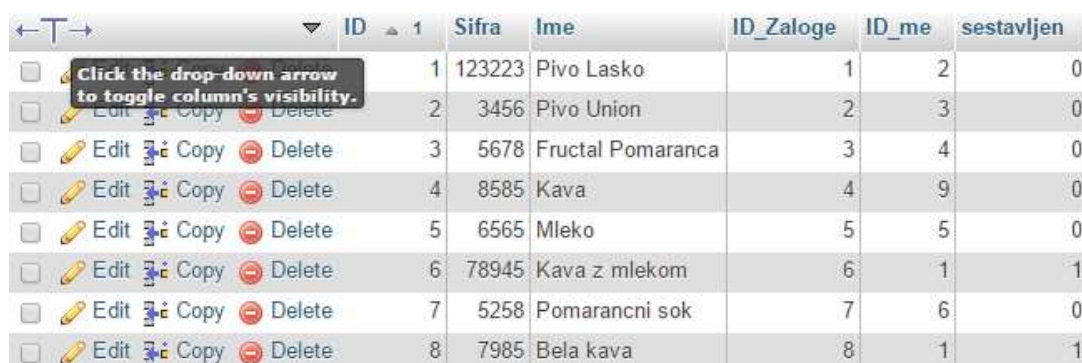
Slika 5: Zaslonska slika tabele *sestavljene*, ki prikazuje dva sestavljena artikla, ki sta sestavljena z večih artiklov.

Tabela 2 prikazuje pomen posameznih stolpcev v tabeli sestavljeni.

Ime stolpca	Razlaga
ID	Enolični identifikator tabele sestavljeni.
ID_Sartikla	Identifikator sestavljenega artikla (kava z mlekom).
ID_zaloge	Identifikator zaloge, ki pripada artiklu, ki je sestavni del sestavljenega artikla, opisanega v stolpcu ID_Sartikla.
količina	Za koliko bomo zmanjšali zalogo artiklu, ki sestavlja sestavljeni artikel opisan v stolpcu ID_Sartikla.

Tabela 2: Pomen stolpcev v tabeli sestavljeni.

Slika 6 prikazuje tabelo *artikel*. V stolpcu *sestavljen* je določeno, ali je artikel sestavljen ali ne.



	ID	Sifra	Ime	ID_Zaloge	ID_me	sestavljen
<input type="checkbox"/>	1	123223	Pivo Lasko	1	2	0
<input type="checkbox"/>	2	3456	Pivo Union	2	3	0
<input type="checkbox"/>	3	5678	Fructal Pomaranca	3	4	0
<input type="checkbox"/>	4	8585	Kava	4	9	0
<input type="checkbox"/>	5	6565	Mleko	5	5	0
<input type="checkbox"/>	6	78945	Kava z mlekom	6	1	1
<input type="checkbox"/>	7	5258	Pomarančni sok	7	6	0
<input type="checkbox"/>	8	7985	Bela kava	8	1	1

Slika 6: Prikaz zaslonska slike tabele *artikel* v podatkovni bazi.

Tabela 3 prikazuje pomen stolpcev v tabeli *artikel*.

Ime stolpca	Razlaga
ID	Enolični identifikator tabele <i>artikel</i> .
Sifra	Šifra črtne kode artikla.
ID_zaloge	Identifikator zaloge, ki pripada artiklu. Nahaja se v tabeli <i>zaloga</i> .
ID_me	Identifikator merske enote, ki se nahaja v tabeli <i>skl_me</i> , ki je šifrant merskih enot artiklov.
Sestavljen	Indikator ali je artikel sestavljen.

Tabela 3: Pomen stolpcev v tabeli *artikel*.

V tabeli *artikel* smo dodali stolpec *sestavljeno*. Če je artikel sestavljen, ima v stolpcu *sestavljeno* vrednost 1, in 0, če ni. Če želimo nadzorovati zalogo artiklov, ki so sestavina sestavljenega artikla, smo morali implementirati način zmanjševanja zalog za te artikle.

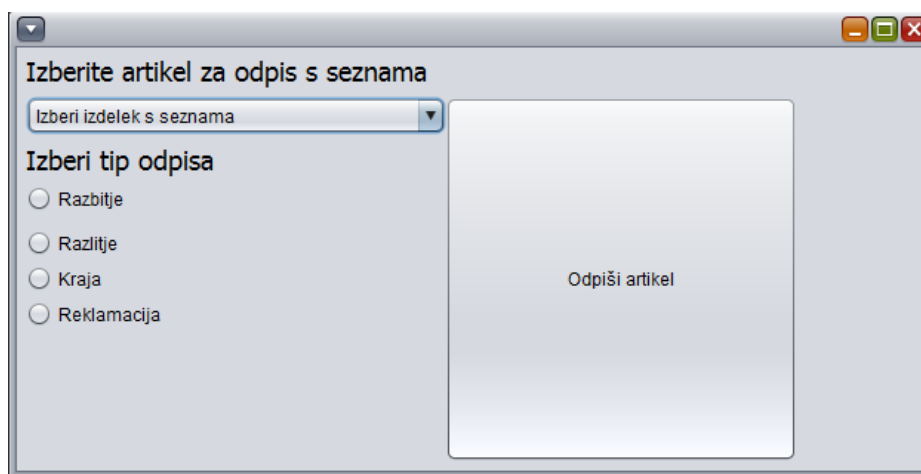
Najprej preverimo, ali je artikel sestavljen. Če ni, zalogo zmanjšamo za 1 kos. Če je artikel sestavljen, ima v stolpcu *sestavljeno* v tabeli *artikel* vrednost 1. V tabeli *sestavljeno* iščemo ID tega sestavljenega artikla, po celotni tabeli. Vsem artiklom, ki so sestavni deli artikla ki ga prodajamo, odštejemo zalogo za količino definirano v stolpcu *količina*, tabele *sestavljeno*. V primeru prodaje kave z mlekom, odštejemo zalogo za 1dl mleka in 5g kave.

Pri modulu *blagajna* smo razvili toliko funkcionalnosti kot jih potrebujemo, da pokažemo optimizacijo procesa v celoti. Tiskanja računov nismo razvijali, saj to ni domena diplomske naloge.

3.4.4 Modul odpis

Ob analizi procesa podjetja smo ugotovili, da je ravnanje z odpisi neučinkovito in časovno potratno. Za beleženje zaloge v realnem času je bilo potrebno implementirati modul za odpis

artiklov. Slika 7 prikazuje modul za odpis artiklov, ki omogoča izbiro artikla ki ga odpisujemo s spustnega seznama. Obvezna je oznaka razloga za odpis.



Slika 7: Modul za odpis artiklov v primerih razbitja, razlitja, kraje ali reklamacije artikla.

Uporaba modula za odpis mora biti preprosta, saj se odpisi dogajajo med samim delovnim procesom. Zaposleni med delovnim procesom nima veliko časa za delo na računalniku, zato mora biti opis opravljen hitro. Ocenili smo, da je za izbiro najučinkovitejša uporaba spustnega seznama. Artikel za odpis izberemo tako, da v sam spustni seznam vpišemo začetnico naziva artikla. Za namene dnevniških in delovnih zapisov nas zanima razlog odpisa, zato smo implementirali označitev razloga odpisa. Za označitev razloga odpisa smo uporabili JRadioButton in jih združili v ButtonGroup, tako, da je možna izbira le enega od razlogov.

Ob kliku na gumb za odpis artikla preverimo, ali je artikel ki se odpisuje sestavljen ali ne. V primeru da artikel ni sestavljen, zmanjšamo zalogo za 1 kos. V primeru da je artikel sestavljen, pa si pomagamo s tabelo *sestavljene* in zmanjšamo zalogo tistim artiklom, kateri sestavljajo artikel, ki ga odpisujemo.

Točene izdelke odpisujemo po različni količini. Lahko se razlije deciliter soka, lahko pa le nekaj kapljic. V ta namen se za te izdelke pojavi številčnica, kjer uporabnik izbere količino, ki jo bo odpisal.

Slika 8 prikazuje modul za odpis artikla v primeru, da odpisujemo artikel, pri katerem se lahko odpiše različna količina artikla. Primer take vrste odpisa je pomarančni sok, ki ga lahko natakara razlije preden ga postreže stranki. Lahko ocenimo in izberemo količino soka, ki se je

razlil. V primeru, da se je razlila količina, katere na številčnici ni prikazane, lahko vpišemo poljubno količino.

Slika 8: Modul za odpis artiklov v primeru odpisa artikla, kjer se lahko odpiše različna količina artikla.

3.4.5 Naročilo artiklov

Odločili smo se, da bomo modul za naročilo artiklov implementirali tako, da bo proces naročila avtomatiziran. Pomembno je, da artikla ob naročilu pri dobavitelju v skladišču ne zmanjka. Slika 9 prikazuje tabelo *zaloga*, v kateri beležimo zalogo posameznih artiklov. Stolpec *min_zaloga* predstavlja mejo za naročilo artikla.

	ID	ID_artikla	zaloga	min_zaloga	max_zaloga	narocilo	ID_me
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	1	260	250	800	100	8
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	2	440	400	500	100	8
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	3	396	200	500	100	4
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	4	4	1	5	2	9
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	5	3	1	5	2	5
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	6	6	0	0	0	0	1
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	7	7	95.8	50	200	100	6
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	8	8	0	0	0	0	1
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	9	9	50	25	100	25	1
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	10	10	50	25	100	25	1
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	11	11	500	300	800	100	1
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	12	12	789	500	800	100	2

Slika 9: Tabela *zaloga* v kateri beležimo zalogo artiklov v skladišču.

Tabela 4 prikazuje pomen posameznih stolpcev v tabeli *zaloga*.

Ime stolpca	Razlaga
ID	Enolični identifikator tabele zaloga.
ID_artikla	Identifikator artikla kateremu pripada zaloga.
Zaloga	Trenutna zaloga artikla v skladišču.
Min_zaloga	Minimalna zaloga artikla v skladišču. Ko zaloga pade pod vrednost minimalne zaloge sprožimo alarm ali artikel avtomatsko naročimo.
Max_zaloga	Maksimalna zaloga artikla v skladišču. Trenutna zaloga ne sme presegati vrednost maksimalne zaloge.

Tabela 4: Razlaga stolpcev v tabeli *zaloga*.

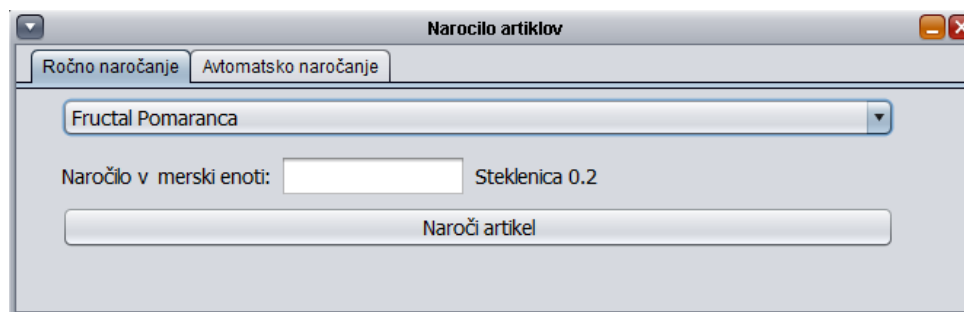
Stolpec *zaloga* nam pove trenutno stanje zaloge v skladišču. Z maksimalno zalogo omejimo največje število artiklov na zalogi. S tem se izognemo nepremišljenemu naročilu artiklov, številu artiklov, ki bi jim utegnil preteči rok uporabe, kopičenju zaloge in posledično prostorski stiski v skladišču.

Stolpec *min_zaloga* je ključen pri avtomatskem naročilu in alarmiranju ob padcu zaloge artikla pod dovoljeno minimalno zalogo. Gre za indikator, pri kateri zalogi se bo sprožil alarm da artikla zmanjkuje, ali pa se bo artikel avtomatsko naročil.

Za upravljanje z zalogo skrbi javanski razred *naročilnik*. Razred smo umestili v modul za odpis in prodajo blaga, saj se v teh dveh modulih zaloga artiklov zmanjšuje. V drugih modulih razred ni potreben, saj ne prihaja do zmanjševanja zaloge.

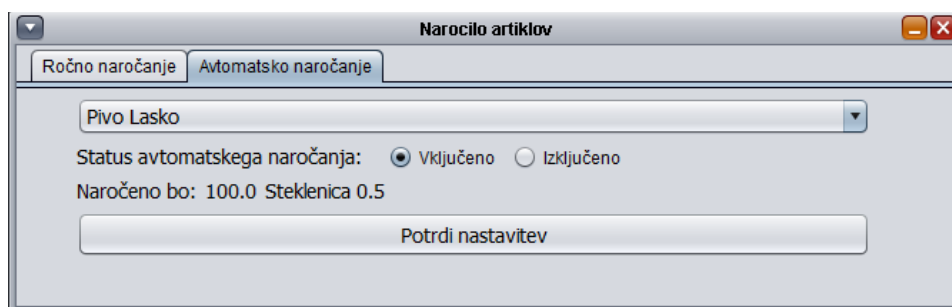
Prototip aplikacije omogoča, da lahko artikle naročamo avtomatsko ali ročno. V primeru, da se artikel naroča avtomatsko, ga ročno ne moremo naročiti.

Slika 10 prikazuje modul v prototipu aplikacije za ročno naročanje artiklov. V tem modulu naročimo poljubno količino artikla s seznama.



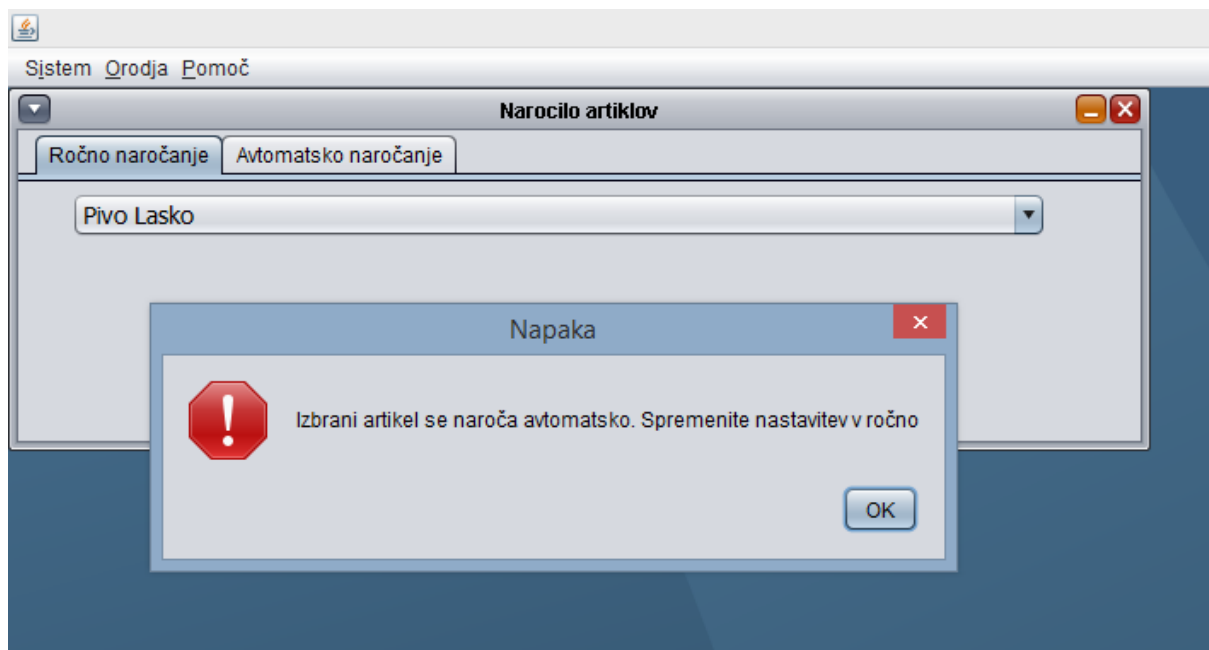
Slika 10: Prikaz zaslonske maske za ročno naročilo artiklov.

Slika 11 prikazuje modul za avtomatko naročanje artiklov. V tem modulu lahko avtomatsko naročanje za posamezen artikel vklapljamo ali izklapljamo.



Slika 11: Prikaz zaslonske maske za avtomatsko naročilo artiklov.

V modulu imamo 2 zavihka. Ročno naročanje in avtomatsko naročanje. Zavihek ročno naročanje nam prikazuje seznam artiklov v skladišču in vnosno polje za vpis količine naročila artikla. Zavihek avtomatsko naročanje nam ponudi spustni seznam in gumba, ki omogočata vklop ali izklop avtomatskega naročanja. Slika 12 prikazuje okno, ki se pojavi v primeru, da je vklopljeno avtomatsko naročilo artiklov, uporabnik pa želi naročiti artikel ročno.



Slika 12: Prikaz napake ob izbiri ročnega naročanja, če je vklopljeno avtomatsko naročilo artiklov.

Razred *naročilnik* ob vsaki izdaji ali odpisu blaga preveri, ali v podatkovni bazi obstaja primer, ko je pri določenem artiklu trenutna zaloga manjša od minimalne zaloge.

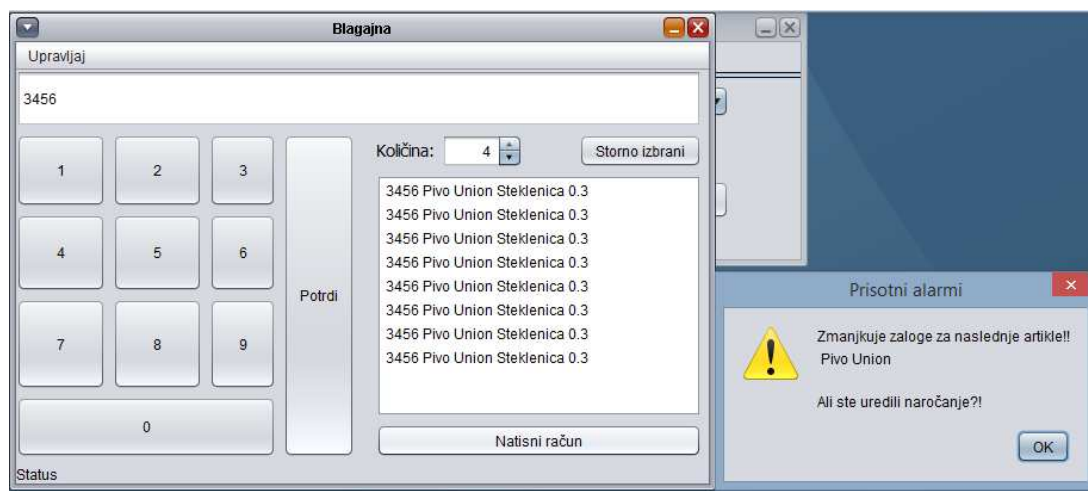
V primeru avtomatskega naročanja se s pomočjo javanskega razreda *sendEmail* in knjižnice *javax.mail* dobavitelju pošlje e-sporočilo, da v lokalu zmanjkuje artikla, za katerega je zadolžen dobavitelj. Pri izdelavi razreda *sendEmail* smo se opirali na dokumentacijo knjižnice *javax.mail*. [12].

Slika 13 prikazuje e-sporočilo ki ga prejme dobavitelj, ko zaloga artikla z vklopljenim avtomatskim naročanjem pade pod minimalno zalogo.



Slika 13: Primer sporočila, ki ga dobavitelj prejme ob zmanjkovanju artikla, z vklopljenim avtomatskim naročanjem.

Slika 14 prikazuje opozorilo, ki se pojavi v primeru zmanjkovanja artikla z vklopljenim ročnim naročanjem.



Slika 14: Prikaz opozorila ob padcu zaloge artikla z vklopljenim ročnim naročanjem pod minimalno zalogo.

Razred *naročilnik* ob vsaki prodaji ali odpisu artikla v podatkovni bazi preveri, ali je zaloga katerega artikla manjša od minimalne zaloge. V primeru, da je bil artikel naročen, sistem dobavitelju več ne pošilja novih naročil.

Slika 15 prikazuje tabelo *pendingorder*. V primeru, da artikla zmanjkuje, vendar je bil naročen, se za artikel indikator *pending* nastavi na 1. Do novega vnosa zaloge v skladišče se sporočila dobaviteljem ne bodo več pošiljala.



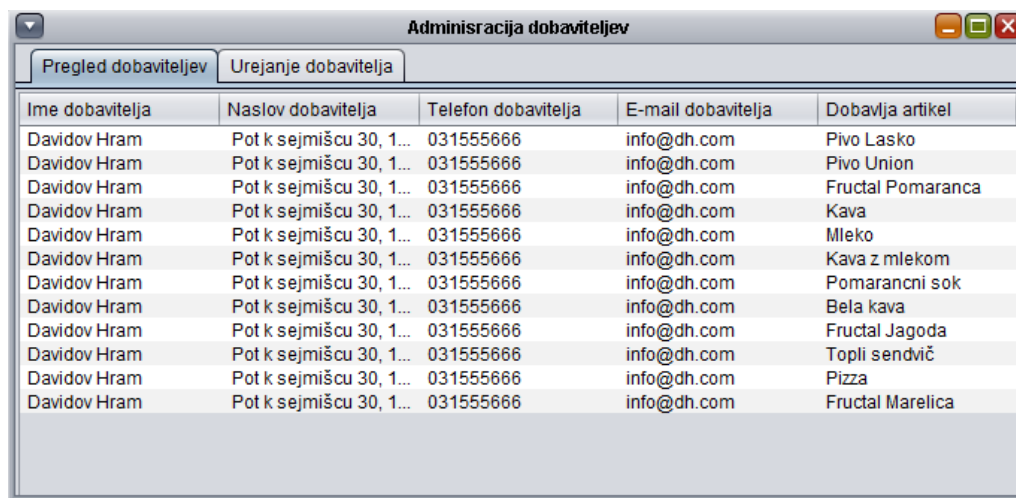
	ID	ID_artikla	pending
Edit Copy Delete	1	1	0
Edit Copy Delete	2	2	1
Edit Copy Delete	3	3	0
Edit Copy Delete	4	4	0
Edit Copy Delete	5	5	0
Edit Copy Delete	6	6	0
Edit Copy Delete	7	7	0
Edit Copy Delete	8	8	0
Edit Copy Delete	9	9	0
Edit Copy Delete	10	10	0
Edit Copy Delete	11	11	0

Slika 15: Prikaz tabele *pendingorder*, ki preprečuje vnovično pošiljanje sporočil dobaviteljem in izpisovanje alarmov do vnosa artiklov v skladišče.

3.4.6 Modul dobavitelji

Dobavitelji so pomemben akter nabavno-prodajnega procesa. Skrbijo, da artiklov pri strankah ne zmanjkuje. V našem prototipu smo omogočili pregled in manipulacijo z dobavitelji artiklov preko modula dobavitelji.

Slika 16 prikazuje zaslonsko masko modula za pregled in urejanje dobaviteljev.



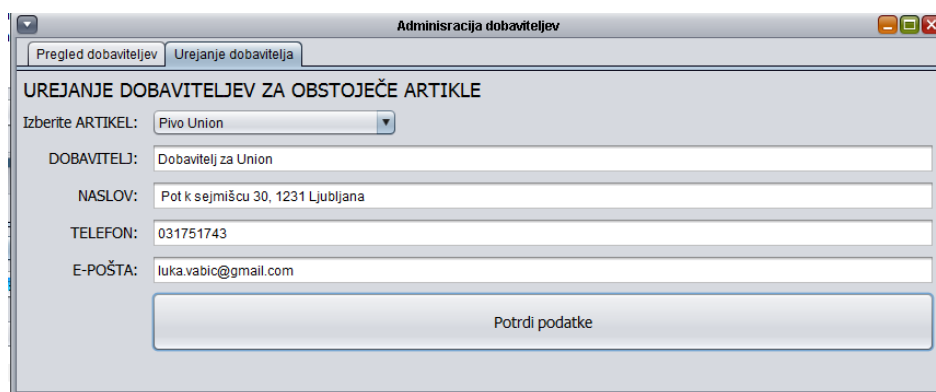
The screenshot shows a window titled 'Administracija dobaviteljev' with two tabs: 'Pregled dobaviteljev' (selected) and 'Urejanje dobavitelja'. Below the tabs is a table with five columns: 'Ime dobavitelja', 'Naslov dobavitelja', 'Telefon dobavitelja', 'E-mail dobavitelja', and 'Dobavlja artikel'. The table contains 11 rows of data, all with the same supplier information but different products.

Ime dobavitelja	Naslov dobavitelja	Telefon dobavitelja	E-mail dobavitelja	Dobavlja artikel
Davidov Hram	Pot k sejmišču 30, 1...	031555666	info@dh.com	Pivo Lasko
Davidov Hram	Pot k sejmišču 30, 1...	031555666	info@dh.com	Pivo Union
Davidov Hram	Pot k sejmišču 30, 1...	031555666	info@dh.com	Fructal Pomaranca
Davidov Hram	Pot k sejmišču 30, 1...	031555666	info@dh.com	Kava
Davidov Hram	Pot k sejmišču 30, 1...	031555666	info@dh.com	Mleko
Davidov Hram	Pot k sejmišču 30, 1...	031555666	info@dh.com	Kava z mlekom
Davidov Hram	Pot k sejmišču 30, 1...	031555666	info@dh.com	Pomarančni sok
Davidov Hram	Pot k sejmišču 30, 1...	031555666	info@dh.com	Bela kava
Davidov Hram	Pot k sejmišču 30, 1...	031555666	info@dh.com	Fructal Jagoda
Davidov Hram	Pot k sejmišču 30, 1...	031555666	info@dh.com	Topli sendvič
Davidov Hram	Pot k sejmišču 30, 1...	031555666	info@dh.com	Pizza
Davidov Hram	Pot k sejmišču 30, 1...	031555666	info@dh.com	Fructal Marelica

Slika 16: Prikaz modula za pregled in administracijo dobaviteljev.

Z modulom pregledujemo dobavitelje za določen artikel, ki jih lahko urejamo. Vsak artikel mora imeti vpisane podatke o dobavitelju. Dobavitelje lahko urejamo samo za nesestavljene artikle.

Slika 17 prikazuje zaslonsko sliko modula za urejanje podatkov o dobaviteljih za posamezen artikel.



The screenshot shows the 'Urejanje dobavitelja' tab in the 'Administracija dobaviteljev' window. The title is 'UREJANJE DOBAVITELJEV ZA OBSTOJEČE ARTIKLE'. There is a dropdown menu for 'Izberite ARTIKEL:' with 'Pivo Union' selected. Below it are text input fields for 'DOBAVITELJ:', 'NASLOV:', 'TELEFON:', and 'E-POŠTA:'. The values entered are 'Dobavitelj za Union', 'Pot k sejmišču 30, 1231 Ljubljana', '031751743', and 'luka.vabic@gmail.com' respectively. At the bottom is a button labeled 'Potrdi podatke'.

Slika 17: Zaslonska slika modula za urejanje dobavitelja za posamezen artikel.

3.4.7 Vnos zaloge artiklov v sistem

Vnos zaloge artiklov je pomemben del optimizacije storitvenega procesa. Procesi, v katerih ne prihaja do beleženja zaloge v realnem času, količinskega vnosa artiklov (ki služi beleženju in manipulaciji z zalogo), ne poznajo.

Za zaposlenega je ključen hiter ter pravilen vnos prihajajočih novih artiklov v skladišče. Ko dobavitelj v skladišče dostavi naročene artikle, prevzemnik podpiše dobavnico, na kateri je zapisano koliko ter katere artikle je stranka naročila.

Slika 18 prikazuje modul za vnos nove zaloge artiklov v sistem.

Slika 18: Prikaz zaslonske slike modula za vnos zaloge artiklov v sistem.

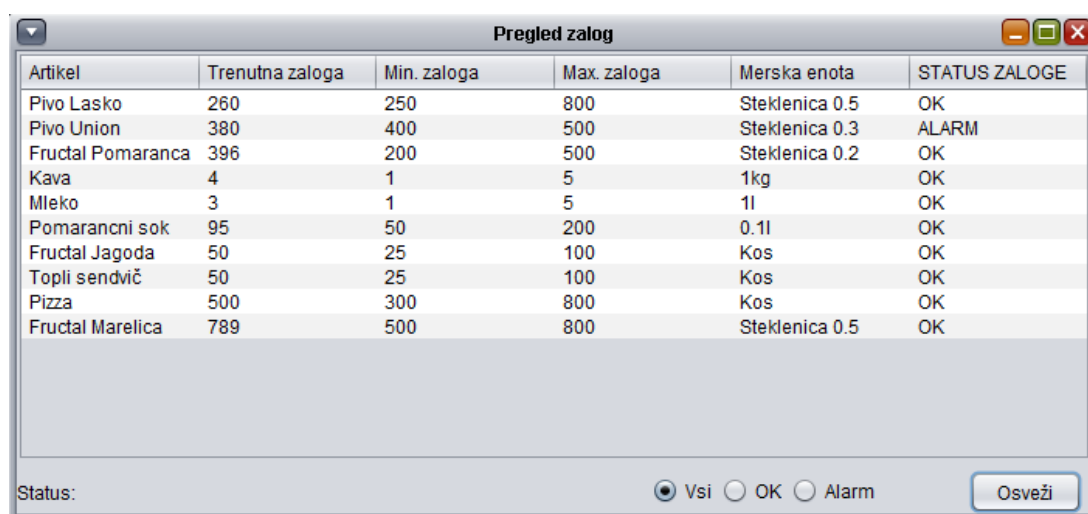
Vnos zaloge artiklov je modul, ki nam ponudi spustni seznam artiklov ter vnosno polje za vnos količine zaloge. Zaloge preko tega modula ne moremo zmanjšati, saj v programski kodi ne dopuščamo vnosa negativnega števila.

Ob vnosu artikla v sistem, se indikator v tabeli *pendingorder* spremeni na 0. Artikel lahko ročno naročimo takoj ali pa se bo naročil sam, ko bo trenutna zaloga manjša od minimalne zaloge.

3.4.8 Modul za pregled zalog in alarmov

Modul pregled zalog nam pokaže stanje zaloge posameznega artikla. Modul preveri, ali je stanje zaloge alarmantno ali ne. Prototip aplikacije omogoča sortiranje po statusu zaloge z radio gumbi. V primeru, da je v skladišču veliko število artiklov in bi želeli pregledati samo tiste, ki jih zmanjkuje, to storimo s klikom na radio gumb Alarm.

Slika 19 nam prikazuje modul za pregled zalog. Prikazani so vsi artikli v skladišču, ne glede na status.

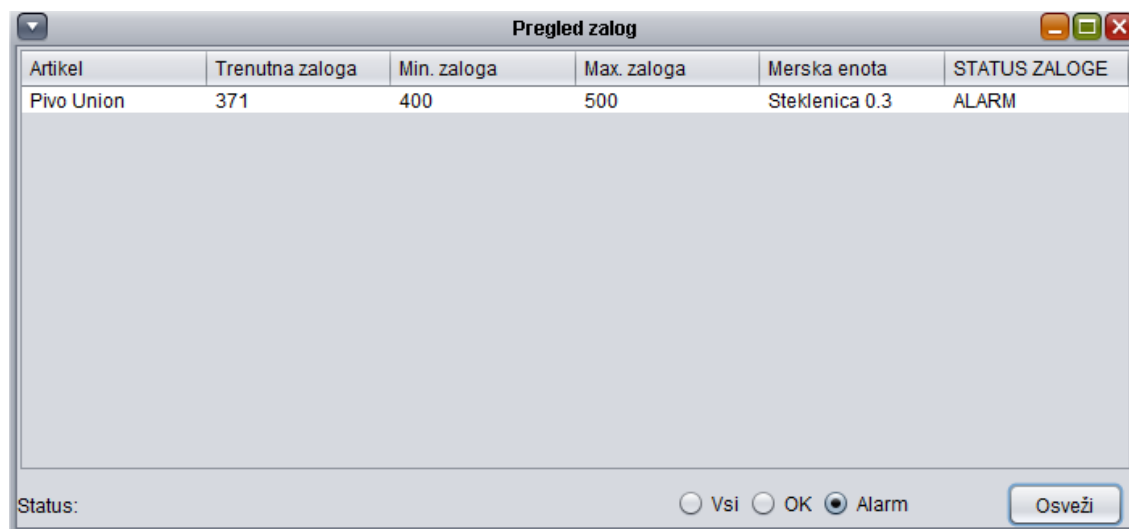


Artikel	Trenutna zaloga	Min. zaloga	Max. zaloga	Merska enota	STATUS ZALOGE
Pivo Lasko	260	250	800	Steklenica 0.5	OK
Pivo Union	380	400	500	Steklenica 0.3	ALARM
Fructal Pomaranca	396	200	500	Steklenica 0.2	OK
Kava	4	1	5	1kg	OK
Mleko	3	1	5	1l	OK
Pomarančni sok	95	50	200	0.1l	OK
Fructal Jagoda	50	25	100	Kos	OK
Topli sendvič	50	25	100	Kos	OK
Pizza	500	300	800	Kos	OK
Fructal Marelica	789	500	800	Steklenica 0.5	OK

Status: ☒ Vsi ☐ OK ☐ Alarm Osveži

Slika 19: Zaslonska slika modula za pregled zalog v sistemu, brez vklopljenih filtrov.

Slika 20 prikazuje modul za pregled zalog. Prikazani so vsi artikli v skladišču, katerim je zaloga padla pod minimalno dovoljeno zalogo.



Artikel	Trenutna zaloga	Min. zaloga	Max. zaloga	Merska enota	STATUS ZALOGE
Pivo Union	371	400	500	Steklenica 0.3	ALARM

Status: ☐ Vsi ☐ OK ☒ Alarm

Slika 20: Zaslonska slika modula za pregled zalog v sistemu, z vklopljenim filtrom *alarm*.

3.4.9 Vnos novega artikla v sistem

V sistem lahko vnesemo nove artikle. Vnosa sestavljenih artiklov nismo predvideli.

Slika 21 prikazuje modul za vnos novih artiklov v sistem.

Slika 21: Zaslonska maska modula za vnos novih artiklov v sistem.

Za vnos novega artikla v sistem je potrebno pravilno vnesti naslednje podatke:

- šifro artikla,
- naziv artikla,
- trenutno zalogo artikla v skladišču,
- dovoljeno minimalno zalogo,
- dovoljeno maksimalno zalogo,
- kolikšno količino artikla bomo naročili ob vklopljenem avtomatskem naročanju,
- ime, naslov, e-pošto in telefon dobavitelja,
- nastavev avtomatskega ali ročnega naročanja.

Ob kliku na gumb za vnos artikla, se v tabele podatkovne baze *artikel*, *zaloga*, *pendingorder*, *dobavitelji* in *auto_order* zapišejo informacije o vnosu novega artikla. Ob kliku na gumb vnos artikla, je nov artikel vnešen v sistem.

3.4.10 Dnevniški in delovni zapisi

V sodobnih logističnih sistemih prihaja do implementacije delovnih zapisov. V angleščini imajo posebno ime, reče se jim audit trail. Njihova značilnost je, da vsebujejo vse sekvence transakcij, ki so se izvajale med uporabo programa. Sledenje je pomembno, saj ne želimo da bi prihajalo do nedovoljenih transakcij kot so neutemeljeni odpisi ali nepotrebna naročila med storitvenim procesom. Z delovnimi zapisi beležimo, kdo in katere spremembe je izvršil v samem programu. Ne gre le za spremljanje sprememb v programu, vendar nam lahko delovni zapisi služijo tudi za beleženje statistike.

V našem prototipu smo združili oboje. Implementirali smo delovne zapise ter jih prilagodili za preprosto beleženje statistike. Slika 22 prikazuje modul za pregled delovnih zapisov v prototipu aplikacije.

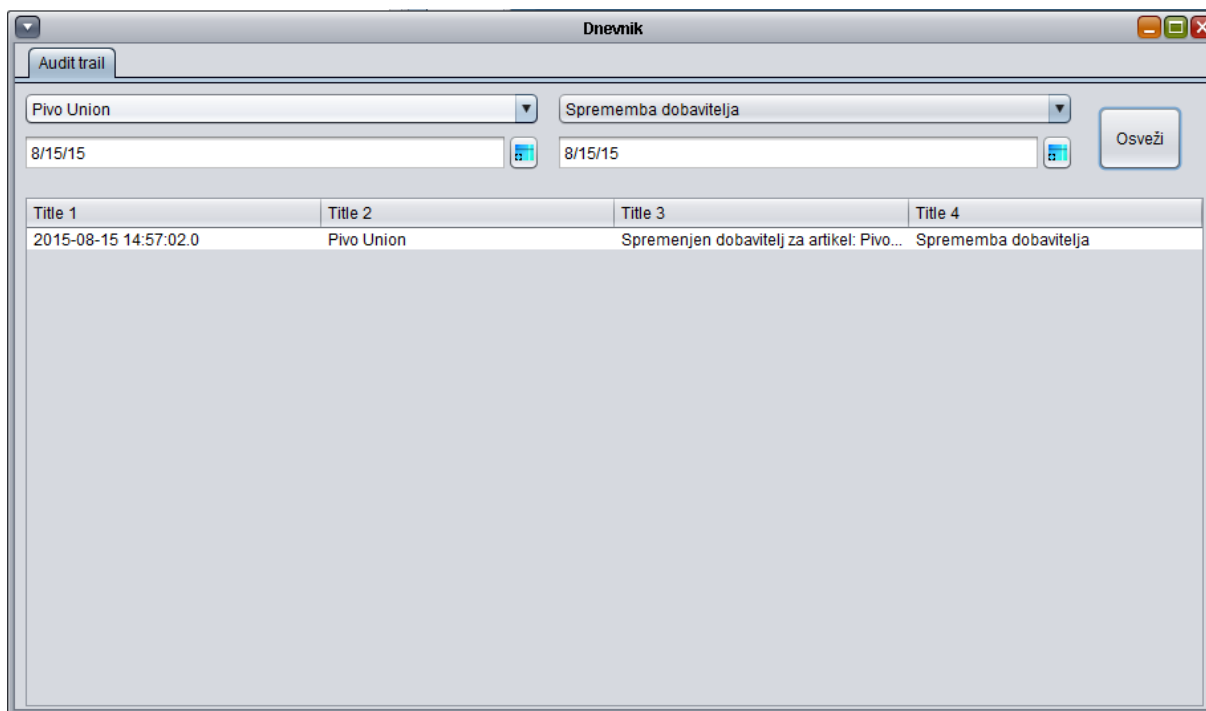
Cas	Artikel	Opis dogodka	Dogodek
2015-08-25 18:23:35.0	Pivo Lasko	Nakup artikla: Pivo Lasko	Nakup
2015-08-25 18:26:05.0	Pivo Union	Nakup artikla: Pivo Union	Nakup
2015-08-25 18:26:10.0	Pivo Lasko	Nakup artikla: Pivo Lasko	Nakup
2015-08-25 19:03:45.0	Pivo Union	Vnos zaloge za: Pivo Union vnos: 50	Vnos nove zaloge
2015-08-25 19:04:10.0	Pivo Union	Spremenjena nastavitve naročanja n...	Spremenjena nastavitve naročanja
2015-08-25 19:13:21.0	Pivo Lasko	Spremenjena nastavitve naročanja n...	Spremenjena nastavitve naročanja
2015-08-25 19:13:24.0	Pivo Lasko	Spremenjena nastavitve naročanja n...	Spremenjena nastavitve naročanja
2015-08-25 19:13:25.0	Pivo Lasko	Spremenjena nastavitve naročanja n...	Spremenjena nastavitve naročanja
2015-08-25 19:14:27.0	Pivo Union	Spremenjena nastavitve naročanja n...	Spremenjena nastavitve naročanja
2015-08-25 19:58:59.0	Pivo Union	Spremenjena nastavitve naročanja n...	Spremenjena nastavitve naročanja
2015-08-25 19:59:13.0	Pivo Union	Ročno naročilo artikla: Pivo Union	Artikel naročen
2015-08-25 20:02:06.0	Pivo Union	Spremenjena nastavitve naročanja n...	Spremenjena nastavitve naročanja
2015-08-25 20:07:17.0	Pivo Union	Spremenjena nastavitve naročanja n...	Spremenjena nastavitve naročanja
2015-08-25 20:07:33.0	Pivo Union	Ročno naročilo artikla: Pivo Union	Artikel naročen
2015-08-25 20:09:23.0	Pivo Lasko	Avtomatsko naročilo artikla: Pivo Lasko	Artikel naročen
2015-08-25 20:09:23.0	Pivo Lasko	Nakup artikla: Pivo Lasko	Nakup
2015-08-25 20:09:57.0	Pivo Lasko	Vnos zaloge za: Pivo Lasko vnos: 20	Vnos nove zaloge
2015-08-25 20:10:22.0	Pivo Lasko	Vnos zaloge za: Pivo Lasko vnos: 50	Vnos nove zaloge
2015-08-25 20:10:32.0	Pivo Union	Vnos zaloge za: Pivo Union vnos: 20	Vnos nove zaloge
2015-08-25 20:19:59.0	Pivo Lasko	Odoisan artikel: Pivo Lasko razlog: R...	Razbitie

Slika 22: Prikaz modula za pregled delovnih zapisov.

Privzeti prikaz je brez omogočenih filtrov. V našem prikazu delovnih zapisov beležimo časovno značko dogodka, ime artikla (saj manipuliramo z artikli), opis dogodka in vrsto dogodka.

Ker v prototipu aplikacije nismo implementirali avtorizacije uporabnikov, stolpec, kdo je izvedel transakcijo, ni prikazan. Pomembno je, da je v delovnem zapisu prikazana informacija o tem, ali je transakcija povzročila spremembo artikla.

Za primer vzemimo posodobitev podatkov dobavitelja. Zapisano mora biti kdaj se je posodobitev zgodila (časovna značka), za kateri artikel smo spremenili dobavitelja ter nove podatke, ki smo jih vpisali. Za lažji pregled in za potrebe statistike, smo v prototipu aplikacije implementirali možnost nastavitve filtriranih podatkov. Slika 23 prikazuje zaslonsko masko modula za pregled delovnih zapisov z vklopljenimi filtri.



Slika 23: Prikaz modula za pregled delovnih zapisov z vkljopljenimi filtri.

Prototip aplikacije omogoča, da si v modulu za pregled delovnih zapisov lahko izberemo naslednje kategorije:

- pregled vseh opravljenih transakcij od začetka delovanja sistema,
- vse transakcije na točno določen dan,
- vse transakcije v določenem obdobju,
- vse transakcije za izbran artikel,
- vse transakcije za izbran artikel na določen dan,
- vse transakcije za artikel v določenem obdobju,
- pregled nakupov za vse artikle,
- pregled razlitij (odpisanih artiklov) za vse artikle,

- pregled razbitij (odpisanih artiklov) za vse artikle,
- pregled reklamacij (odpisanih artiklov) za vse artikle,
- pregled kraj (odpisanih artiklov) za vse artikle,
- pregled pojavitve alarma za vse artikle,
- pregled izvedbe naročila za vse artikle,
- pregled vnosov nove zaloge za vse artikle,
- pregled sprememb nastavitev naročanja za vse artikle,
- pregled sprememb dobavitelja za vse artikle,
- pregled nakupov za izbran artikel,
- pregled razlitij (odpisanih artiklov) za izbran artikel,
- pregled razbitij (odpisanih artiklov) za izbran artikel,
- pregled reklamacij (odpisanih artiklov) za izbran artikel,
- pregled kraj (odpisanih artiklov) za izbran artikel,
- pregled pojavitve alarma za izbran artikel,
- pregled izvedbe naročila za izbran artikel,
- pregled vnosov nove zaloge za izbran artikel,
- pregled sprememb nastavitev naročanja za izbran artikel,
- pregled sprememb dobavitelja za izbran artikel,
- pregled nakupov za izbran artikel na določen datum,
- pregled razlitij za izbran artikel na določen datum,
- pregled razbitij za izbran artikel na določen datum,
- pregled reklamacij za izbran artikel na določen datum,
- pregled kraj za izbran artikel na določen datum,
- pregled pojavitve alarma za izbran artikel na določen datum,
- pregled izvedbe naročila za izbran artikel na določen datum,
- pregled vnosov nove zaloge za izbran artikel na določen datum,
- pregled sprememb nastavitev naročanja za izbran artikel na določen datum,
- pregled sprememb dobavitelja za izbran artikel na določen datum,
- pregled nakupov za izbran artikel v določenem obdobju,
- pregled razlitij za izbran artikel v določenem obdobju,
- pregled razbitij za izbran artikel v določenem obdobju,

- pregled reklamacij za izbran artikel v določenem obdobju ,
- pregled kraj za izbran artikel na določen datum v določenem obdobju,
- pregled pojavitve alarma za izbran artikel v določenem obdobju,
- pregled izvedbe naročila za izbran artikel v določenem obdobju,
- pregled vnosov nove zaloge za izbran artikel v določenem obdobju,
- pregled sprememb nastavitvev naročanja za izbran artikel v določenem obdobju,
- pregled sprememb dobavitelja za izbran artikel v določenem obdobju.

















































































Za beleženje delovnih zapisov smo v razredu *Dbconnect* implementirali metodo *lukaLogger*, ki kot attribute prejme *IDartikla*, *opis*, *transakcija*, in *IDeventa*, ki je šifrant transakcij (nakup, odpis, naročilo...). Metoda *lukaLogger* je klicana v metodah drugih razredov, ki opravljajo transakcije nad celotnim programom (modul blagajna, odpis, naročilo artiklov, vnos zaloge artiklov) in preko nje vstavlja podatke v tabelo logs, s katere modul dnevniški in delovni zapisi črpa podatke in prikaže vsebino na zaslon. Slika 24 prikazuje zaslonsko sliko tabele transakcije v podatkovni bazi prototipa aplikacije.



	ID	event
<input type="checkbox"/> Edit Copy Delete	1	Nakup
<input type="checkbox"/> Edit Copy Delete	2	Razbitje
<input type="checkbox"/> Edit Copy Delete	3	Razlitje
<input type="checkbox"/> Edit Copy Delete	4	Reklamacija
<input type="checkbox"/> Edit Copy Delete	5	Kraja
<input type="checkbox"/> Edit Copy Delete	6	Alarm na artiklu
<input type="checkbox"/> Edit Copy Delete	7	Artikel naročen
<input type="checkbox"/> Edit Copy Delete	8	Vnos nove zaloge
<input type="checkbox"/> Edit Copy Delete	9	Spremenjena nastavitvev naročanja
<input type="checkbox"/> Edit Copy Delete	10	Sprememba dobavitelja

Slika 24: Prikaz zaslonske slike tabele *transakcije* v podatkovni bazi prototipa aplikacije.

Slika 25 prikazuje zaslonsko masko tabele *logs* v podatkovni bazi prototipa aplikacije.

← T →		ID	datetime	ID_artikla	description	ID_event	
	 Edit	 Copy	 Delete	1	2015-07-07 20:22:09	1 Spremenjena nastavitvev naročanja na artiklu Pivo L...	9
	 Edit	 Copy	 Delete	2	2015-07-07 20:22:16	1 Spremenjena nastavitvev naro?anja na artiklu Pivo L...	9
	 Edit	 Copy	 Delete	3	2015-07-07 21:14:39	7 Nakup artikla: Pomarančni sok	
	 Edit	 Copy	 Delete	4	2015-07-07 21:18:27	1 Vnos zaloge za: Pivo Lasko vnos: 20	8
	 Edit	 Copy	 Delete	5	2015-07-07 21:19:15	1 Odpisan artikel: Pivo Lasko razlog: Razbitje	2
	 Edit	 Copy	 Delete	6	2015-07-07 21:19:31	1 Odpisan artikel: Pivo Lasko razlog: Razlitje	3
	 Edit	 Copy	 Delete	7	2015-07-07 21:19:50	1 Spremenjen dobavitelj za artikel: Pivo Lasko	10
	 Edit	 Copy	 Delete	8	2015-07-07 21:20:03	1 Spremenjen dobavitelj za artikel: Pivo Union	10
	 Edit	 Copy	 Delete	9	2015-07-08 15:00:25	1 Spremenjen dobavitelj za artikel: Pivo Lasko	10
	 Edit	 Copy	 Delete	10	2015-07-08 15:01:39	1 Odpisan artikel: Izberi izdelek s seznama razlog: ...	0
	 Edit	 Copy	 Delete	11	2015-07-08 15:07:25	1 Spremenjen dobavitelj za artikel: Pivo Lasko	10
	 Edit	 Copy	 Delete	12	2015-07-08 15:07:38	2 Spremenjena nastavitvev naro?anja na artiklu Pivo U...	9
	 Edit	 Copy	 Delete	13	2015-07-08 15:08:09	2 Spremenjena nastavitvev naro?anja na artiklu Pivo U...	9
	 Edit	 Copy	 Delete	14	2015-07-08 21:00:24	1 Odpisan artikel: Pivo Lasko razlog: Razbitje	2
	 Edit	 Copy	 Delete	15	2015-08-15 14:41:24	2 Nakup artikla: Pivo Union	1
	 Edit	 Copy	 Delete	16	2015-08-15 14:42:41	2 Spremenjena nastavitvev naro?anja na artiklu Pivo U...	9
	 Edit	 Copy	 Delete	17	2015-08-15 14:43:42	2 Nakup artikla: Pivo Union	1
	 Edit	 Copy	 Delete	18	2015-08-15 14:43:55	2 Spremenjena nastavitvev naro?anja na artiklu Pivo U...	9
	 Edit	 Copy	 Delete	19	2015-08-15 14:43:56	2 Spremenjena nastavitvev naro?anja na artiklu Pivo U...	9
	 Edit	 Copy	 Delete	20	2015-08-15 14:44:20	2 Nakup artikla: Pivo Union	1

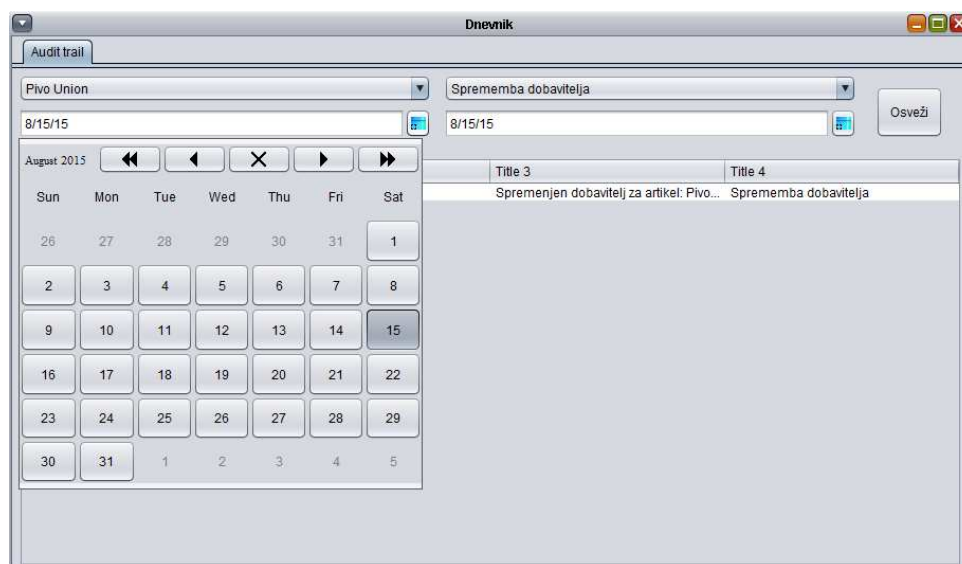
Slika 25: Prikaz zasronske slike tabele *logs* v podatkovni bazi prototipa aplikacije.

Tabela 5 prikazuje pomen posameznih stolpcev v tabeli *logs*.

Ime stolpca	Razlaga
ID	Enolični identifikator tabele logs.
ID_artikla	ID_artikla s tabele artikel. Na artiklu s tem ID jem opravljamo transakcijo.
Description	Opis transakcije.
ID_event	ID dogodka, opisan v šifrantu transakcije.

Tabela 5: Pomen stolpcev v tabeli *logs*.

Za izbiro datuma smo uporabili razred JDate chooser in ga implementirali v prototip za bolj prijetno uporabniško izkušnjo. Za pomoč pri implementaciji smo se opirali na dokumentacijo razreda [13]. Slika 26 prikazuje zaslonsko masko modula za pregled delovnih zapisov ob izbiri datuma, za prikaz katerega smo uporabili javanski razred JDateChooser.



Slika 26: Prikaz zaslonske maske delovnih zapisov ob izbiri datuma.

Za potrebe beleženja statistike smo poleg filtriranja podatkov omogočili izvoz podatkov v formatu .xlsx. Za implementacijo izvoza podatkov smo uporabili API Apache POI [14] in se pri implementaciji opirali na dokumentacijo omenjenega API-ja [15].

Ob kliku na gumb izvoz v excel se odpre okno za izbiro mesta shranjevanja. Datoteka v formatu .xlsx se shrani na izbrano mesto. Ob izvozu filtriranih podatkov lahko opravimo natančne statistične meritve v programu kot je Microsoft Excel. Ocenili smo, da je za beleženje statistike bolje implementirati možnost izvoza podatkov, kot implemenitati sistem za beleženje statistike v samem prototipu aplikacije. Orodja kot so Microsoft Excel in druga (Google Sheets, LibreOffice), so razvita prav za namen močnih kalkulacij in opravljanje analiz. V primeru razvoja sistema za beleženje statistike v prototipu aplikacije, ne bi dosegli želenega standarda obdelave podatkov.

Poglavje 4 Ugotovitve

Z razvojem prototipa aplikacije smo poskušali zadostiti vsem štirim ravnam logističnih odločitev:

- operativni ravni – obdelavi naročil in beleženju stanja zalog itd.,
- kontroli procesov – nadzoru nad prevzemom in izdajo izdelkov v skladišče,
- taktičnemu izvajanju – spremljanju gibanje zaloge,
- strateškemu načrtovanju – ocenjevanju doseženih rezultatov na področju logistike (beleženje statistike in delovnih zapisov) [1].

Glede na analiziran poslovni proces na začetku diplomske naloge, smo poskušali poiskati rešitve za izzive, s katerimi se izbrani gostinski lokal spopada.

V prototipu aplikacije smo implementirali:

- modul za vnos artiklov – tako v vsakem trenutku vemo, koliko artiklov je v skladišču gostinskega lokala;
- modul za odpis blaga – zaloge artiklov se ažurno zmanjšuje in ne prihaja do neskladij zaradi odpisanih artiklov;
- modul za naročilo artiklov – standardno se artikli naročajo avtomatsko, vodja lokala in zaposleni s tem nimajo dodatnega dela;

V primeru, da je pričakovan višek obiska, lahko vodja lokala artikle v sistemu naroči ročno. Nastavitev za avtomatsko ali ročno naročanje izdelkov lahko poljubno vklopi ali izklopi.

- modul za beleženje transakcij, sprememb v programu in statistiko.

Ponovimo izračun, ki smo ga opravili v analizi storitveno poslovnega procesa:

$$4 \times (3 + \widetilde{3.5} + 1 + 1) = \frac{\widetilde{34}}{4} = \widetilde{8.5}$$

Pred optimizacijo ugotovimo, da se za urejanje stanja v skladišču, preverjanje prodaje in vnos artiklov, porabi več kot en delovni dan na teden.

Po izdelanem prototipu lahko ocenimo, za koliko smo optimizirali poslovni proces podjetja:

- inventura se izvaja enkrat mesečno po tri ure in pol,
- za vnos zaloge v skladišče in prepis z dobavnice – največ 15 minut tedensko,
- urejanje odpisov s preverjanjem statistike pa zahteva največ 5 minut tedensko.

Ocenjujemo, da bomo za pregled zalog v skladišču, preverjanje števila prodanih artiklov in vnos artiklov porabili le še okoli 5 ur na mesec.

Zavedamo se, da bi bila za podrobne meritve potrebna implementacija prototipa na živem sistemu, vendar lahko z veliko verjetnostjo napovemo, kaj bi podjetje ob implementaciji našega programa pridobilo. Pridobilo bi:

- beleženje dogodkov v programu;
 - avtomatizirano naročilo artiklov;
 - možnost beleženja zaloge v realnem času;
 - ob odpisu artiklov je poznan razlog;
 - ob odpisu sestavljenih artiklov se pravilno zmanjša količina artiklov, ki jih sestavljajo;
 - zaradi vpeljanih izboljšav z modulom za odpis artiklov je beleženje zaloge v realnem času bolj točno. To pomeni, da je število potrebnih inventur na mesec nižje.
- Ocenjujemo, da bo opravljanje inventure enkrat na mesec zadoščalo potrebam uspešnega poslovanja gostinskega lokala.

Ob razvoju prototipa aplikacije smo naleteli na pomanjkljivosti sistema:

- kljub našim prizadevanjem proces inventure ne bo povsem avtomatiziran, zaradi potrebe po implementaciji strojne rešitve;
- ob nedosledni uporabi prototipa aplikacije, (kar pomeni, da zaposleni v gostinskem lokalu ne vnaša natančno prodanih ali odpisanih artiklov ali slabo oceni količino porabljenih artiklov pri sestavljenih artiklih), lahko prihaja do neskladij z dejanskim stanjem zaloge artiklov v skladišču;
- sistem ne predvideva zlorab sistema s strani zaposlenih (kraja artiklov s skladišča, goljufije pri prodaji artiklov).

Prototip aplikacije bi lahko z manjšimi prilagoditvami implementirali v več dejavnostih kot na primer: manjše obrtne dejavnosti (mizarstvo, krovstvo, žlebarstvo, železnine), trgovine z gradbenim materialom, trgovine z obutvijo in tekstilom in podobno.

4.1 Izzivi ob vpeljavi optimizacije prodajno-nabavnih tokov

Odločitev za optimizacijo procesa je težka. Udeležence v procesu, ki ga želimo optimizirati je potrebno prepričati, da je optimizacija koristna za njih ter podjetje in jim bo olajšala delo. Ta naloga je težka posebej v manjših podjetjih, zaposleni so navajeni procesa in jim ustaljeno delo ne predstavlja težav. Tega se zaveda tudi vodstveni kader, ki se odloča o uvedbi optimizacije. Pomembno je, da mu predstavimo konkretne prednosti in izboljšave, ki bi jih želeli vpeljati.

Težava se pojavi tudi pri samem načrtovanju sistema. Procesa opravljanja inventure ni mogoče v celoti avtomatizirati. Ocenili smo, da bi za rešitev tega problema potrebovali strojno rešitev. Zamislili smo si tehniko, ki bi v realnem času odčitavala težo embalaže in podatke pošiljala naši aplikaciji. Aplikacija bi vrednosti zaloge pretvarjala v ustrezno količinsko enoto. Tako bi zaposleni natanko vedeli, kolikšna je zaloga posameznega artikla v lokalu. S programsko rešitvijo smo se približali želenim rezultatom, vendar nabavno prodajnega procesa nismo uspeli v celoti avtomatizirati.

Za točno manipulacijo zaloge in pravilno delovanje, aplikacija zahteva točen vnos podatkov. Tipičen primer je zmanjševanje zaloge pri artiklih, ki so sestavljeni iz več artiklov. V primeru netočnega vnosa vrednosti, se lahko pojavijo znatne razlike med zalogo artiklov in dejanskim stanjem zaloge artiklov v skladišču. Pomembno je, da se uporabniki aplikacije zavedajo pomena točnosti vnosa podatkov.

Poglavje 5 Sklep

Ugotovimo lahko, da je za optimizacijo nabavno-prodajnih procesov potrebno dobro poznavanje želja in potreb samih udeležencev v procesu, katerega optimiziramo. V samem procesu optimizacije je zaželena stalna komunikacija z udeleženci v procesu. To nam omogoča, da v sistem sproti vnašamo izboljšave in ga prilagajamo zahtevam uporabnika. Ali se optimizacija nabavno-prodajnega procesa finančno izplača, pokažejo šele daljša testiranja in dejanska uporaba izdelanega prototipa aplikacije.

Ob razvijanju rešitve ne smemo biti osredotočeni le na njen programski del. Že ob analizi procesa je potrebno premisliti, ali bomo za popolno optimizacijo procesa morali implementirati tudi strojno rešitev. To nas vodi k zaključku, da je za optimizacijo procesa potrebna ekipa z izkušnjami v programskem razvoju, strojnem razvoju in projektnem vodenju.

Kljub temu, da je logistika relativno mlada veda in se z optimizacijo procesov s pomočjo računalništva, informatike in strojništva srečujemo šele od devetdesetih let dvajsetega stoletja, je potrebno spodbujati miselnost o tem, da je procese, ki kljub razvoju ostajajo enaki že vrsto let, mogoče z obstoječim znanjem o logistiki in tehnologiji optimizirati tudi v manjših proizvodnih in storitvenih dejavnostih.

Literatura

- [1] Vekoslav Potočnik, "Nabavno poslovanje," in *Nabavno poslovanje*. Ljubljana: Ekonomska fakulteta v Ljubljani, enota za založništvo in Inštitut za trženje pri Ekonomski fakulteti, 2002.
- [2] Tomaž Dobravec. (- -K)ava. [Online]. <http://lalg.fri.uni-lj.si/P2/kava/Kava.html#QQ2-2-1>
- [3] (2015, July) Wikipedia. [Online]. <https://sl.wikipedia.org/wiki/MySQL>
- [4] MySQL. Top Reasons to Use MySQL. [Online]. <https://www.mysql.com/why-mysql/topreasons.html>
- [5] Oracle. Java™ SE Development Kit 8, Update 25. [Online]. <http://www.oracle.com/technetwork/java/javase/8u25-relnotes-2296185.html>
- [6] Apache Friends. (2015) XAMPP Apache + MySQL + PHP + Perl. [Online]. <https://www.apachefriends.org/about.html>
- [7] Oracle Corporation. (2015) NetBeans IDE. [Online]. <https://netbeans.org/features/platform/features.html>
- [8] phpMyAdmin. [Online]. <https://www.phpmyadmin.net/>
- [9] Class JDesktopPane. [Online]. <http://docs.oracle.com/javase/7/docs/api/javax/swing/JDesktopPane.html>
- [10] (2015, July) Multiple document interface. [Online]. https://en.wikipedia.org/wiki/Multiple_document_interface
- [11] MySQL. Connecting to MySQL Using the JDBC DriverManager Interface. [Online]. <http://dev.mysql.com/doc/connector-j/en/connector-j-usagenotes-connect-drivermanager.html>
- [12] Sun Microsystems. (2007) Oracle Docs. [Online]. <https://docs.oracle.com/javaee/5/api/javax/mail/package-summary.html>

- [13] Space Science and Engineering Center. (2015, May) Class JDateChooser. [Online]. <http://www.ssec.wisc.edu/mcidas/software/v/javadoc/1.4/edu/wisc/ssec/mcidasv/data/dateChooser/JDateChooser.html>
- [14] The Apache Software Foundation. Apache POI - the Java API for Microsoft Documents. [Online]. <https://poi.apache.org/>
- [15] The Apache Software Foundation. (2015) POI API Documentation. [Online]. <https://poi.apache.org/apidocs/index.html>